

**LAPORAN PRAKTIKUM 1**  
**PEMROGRAMAN LANJUT**  
**BANGUN DATAR DAN BANGUN RUANG**



Oleh

Nama : Rizqillah  
NIM : 1957301020  
Kelas : TI 2C  
Dosen Pembimbing : Musta'inul Abdi, SST., M.Kom.

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**JURUSAN TEKNOLOGI INFORMASI DAN KOMPUTER**  
**TAHUN 2021**

## **LEMBAR PENGESAHAN**

No. Praktikum : 01/PPL/2C/TI/2021  
Judul : Bangun Datar dan Bangun Ruang  
Nama : Rizqillah  
NIM / Kelas : 1957301020 / TI 2C  
Jurusan : Teknologi Informasi Dan Komputer  
Prodi : Teknik Informatika  
Tanggal praktikum : 25 Februari 2021  
Tanggal penyerahan : 03 Maret 2021  
Nilai :

Buketrata, 03 Maret 2021

Dosen Pembimbing,

Musta'inul Abdi, SST., M.Kom.  
NIP. 19911030 20190310 1 5

## DAFTAR ISI

LEMBAR PENGESAHAN .....	i
DAFTAR ISI .....	ii
BAB 1 Pendahuluan.....	1
1.1 Tujuan.....	1
1.2 Dasar Teori .....	1
BAB 2 Pembahasan.....	2
2.1 Membuat Program Bangun Ruang dan Bangun Datar .....	2
2.1.1 Bangun Ruang.....	2
2.1.2 Bangun Datar .....	2
2.2 Program .....	3
2.2.1 Bangun Ruang.....	3
Kerucut.....	3
2.2.2 Bangun Datar .....	9
Lingkaran .....	9
BAB 3 Kesimpulan .....	14
DAFTAR PUSTAKA .....	15

# **BAB 1**

## **PENDAHULUAN**

### **1.1 Tujuan**

Pada sesi ini, kita akan mengantarkan beberapa konsep dasar dari Object-Oriented objects, dan Programming (OOP). Selanjutnya kita akan membahas konsep dari class dan bagaimana menggunakan class dan anggotanya. Perubahan dan pemilihan object juga akan dibahas. Sekarang, kita akan focus dalam menggunakan class yang telah dijabarkan dalam Java Class library, kita akan membahas nanti tentang bagaimana membikin class anda sendiri.

Pada akhir pelajaran, siswa seharusnya dapat :

1. menjelaskan OOP dan beberapa konsepnya
2. perbedaan antara class dan object
3. perbedaan antara instance variables/method dan class (static) variable/method
4. menjelaskan method apa dan bagaimana memanggil method parameter
5. mengidentifikasi beberapa jangkauan dari sebuah variable
6. memilih tipe data primitive dan object
7. membandingkan objects dan menjabarkan class dari objects.

### **1.2 Dasar Teori**

OOP (Object Oriented Programming) adalah suatu metode pemrograman yang berorientasi kepada objek. Tujuan dari OOP diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari. Setiap bagian dari suatu permasalahan adalah objek, objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil lagi. Saya ambil contoh Sepeda Motor, pada sepeda motor pasti memiliki bagian-bagian dari yang terbesar sampai bagian terkecil. Dan bagian-bagian inilah yang disebut objek-objek yang memiliki class-class nya masing-masing. Dan setiap class-class tersebut bisa saja memiliki method yang banyak ataupun method yang sedikit. Setiap method itu bisa kita gunakan agar melakukan pekerjaan yang diinginkan, yaitu seperti jika kita membuat method pada lampu sepeda motor agar hidup, maka jika kita memanggil method tersebut, maka lampu sepeda motor tersebut akan hidup.

Begitu juga dengan program, sebuah objek yang besar dibentuk dari beberapa objek yang lebih kecil, objek-objek itu saling berkomunikasi, dan saling berkiriman pesan kepada objek yang lain.

## **BAB 2**

### **PEMBAHASAN**

#### **2.1 Membuat Program Bangun Ruang dan Bangun Datar**

Dalam praktikum kali ini, kita akan membahas program yang telah dibuat selama praktikum berlangsung, yaitu program bangun ruang dan bangun datar.

##### **2.1.1 Bangun Ruang**

Bangun ruang merupakan salah satu objek matematika yang mempelajari mengenai bangun tiga dimensi.

Jenis-jenis bangun ruang :

1. Kubus
2. Balok
3. Tabung
4. Kerucut
5. Limas Segitiga
6. Limas Segiempat
7. Prisma
8. Bola

##### **2.1.2 Bangun Datar**

Bangun datar adalah suatu bidang datar yang tersusun oleh titik atau garis-garis yang menyatu membentuk bangun 2 dimensi yang mempunyai keliling dan luas. Bangun datar merupakan sebuah aksioma di bidang ilmu matematika khususnya geometri analitik, karena hal ini dapat terbukti dengan sendirinya tanpa melakukan pembuktian matematika lebih lanjut [H.S.M. Coxeter, "*Introduction to geometry*", Wiley (1969) pp. 178]. Dalam bahasa inggris bangun datar disebut dengan *plane geometry*.

Jenis-jenis bangun datar :

1. Segitiga
2. Lingkaran
3. Persegi
4. Persegi Panjang
5. Jajar Genjang
6. Trapesium
7. Belah Ketupat
8. Layang-layang

## 2.2 Program

### 2.2.1 Bangun Ruang

#### ▪ Kerucut

##### Coding :

```
package bangunRuang;
import java.util.Scanner;

//Membuat class KerucutRZQ
//Serta membuat atribut yang diperlukan nantinya
public class KerucutRZQ {
    private final double pi = 3.14;
    private double jari, tinggi, s, volume, luas;
    Scanner input = new Scanner(System.in);

    public KerucutRZQ(){
        //Memanggil method show()
        show();
        //variabel method construct
        int pilihan;
        boolean ulang = true;

        //melakukan perulangan jika nilai ulang masih true
        while(ulang){

            //melakukan inputan pilihan
            System.out.print("Kerucut\nAnda ingin
Menghitung?\n1.Volume\n2.Luas Selimut\n3.Luas Alas\n4.Luas
Permukaan"
                                + "\n5.Cari Nilai Garis Pelukis\n\nMasukkan
Input anda : ");
            pilihan = input.nextInt();

            //mengecek apakah pilihan yang dipilih antara 1
sampai 5
            if(pilihan == 1 || pilihan == 2 || pilihan == 3 ||
pilihan == 4 || pilihan == 5){

                //jika pilihan benar, maka akan diminta input
jari-jari yang diketahui dan akan memanggil method setJari
                //untuk diisi nilai jari-jari melalui parameter
```

```

        System.out.print("\nMasukkan jumlah nilai Jari-
jari (cm)\t\t: ");
        this.jari = input.nextDouble();
        setJari(this.jari);

        //cek apakah pilihan == 1 atau 5, jika iya
diminta input tinggi
        if(pilihan == 1 || pilihan == 5){
            System.out.print("Masukkan jumlah nilai
Tinggi (cm)\t\t: ");
            this.tinggi = input.nextDouble();
            setTinggi(this.tinggi);

            //cek apakah pilihan == 2 atau 4, jika iya
diminta input garis pelukis
            //atau nilai yang nantinya diisi ke variabel s
        } else if(pilihan == 2 || pilihan == 4){
            System.out.print("Masukkan jumlah nilai
Garis Pelukis (cm)\t: ");
            tinggi = input.nextDouble();
            setGarisPelukis(this.tinggi);
        }

        //apabila pilihan yang dimasukkan sama dengan 1
maka akan dipanggil method volume, agar dicari berapa volume
        if(pilihan == 1){
            Volume();
            //dan jika pilihan 2, maka akan dipanggil
method LuasSelimut
        } else if(pilihan == 2) {
            LuasSelimut();
        } else if(pilihan == 3){
            LuasAlas();
        } else if(pilihan == 4){
            LuasPermukaan();
        } else {
            //jika semua if nilai false, maka akan dicetak
nilai garis pelukis
            System.out.println("Nilai Garis Pelukis
adalah = " + Math.round(getGarisPelukis()) + " cm\n");
        }

```

```

        //pada sintaks dibawah ini untuk bertanya
        kepada user ingin keluar ata tidak, pilihan yang bisa
        dimasukkan

        //hanyalah huruf (y,Y,yes,Yes,n,N,no,No),
        selain dari itu akan muncul tidak valid
        System.out.print("Anda ingin Keluar?? (Y/N)");
        String result = input.next();

        //mengecek apakah yang diinput Y atau y dan
        lain-lain

        //maka akan menghasilkan nilai ulang = false,
        yang berarti keluar dari while
        switch (result) {
            case "Y":
            case "y":
            case "yes":
            case "Yes":
                ulang = false;
                break;
            //mengecek apakah yang diinput N atau n dan
            lain-lain

            //maka akan menghasilkan nilai ulang =
            true, dan berarti perulangan while akan tetap berjalan
            case "N":
            case "n":
            case "no":
            case "No":
                ulang = true;
                break;
            //dan jika yang dimasukkan adalah lainnya,
            maka akan mencetak Inputan SALAH!!
            //dan memberi nilai false pada variabel
            ulang, agar perulangan while berhenti
            default:
                System.out.println("Inputan SALAH!!");
                ulang = false;
                break;
        }

        System.out.print("\n=====
        ===\n\n");

```



```

        //dan jika pilihan yang dimasukkan bukan angka 1,
        2, 3, 4, atau 5, maka akan mencetak pilihan anda tidak valid
        //dan memberi nilai false pada variabel ulang, agar
        keluar dari while
    } else {
        System.out.print("Pilihan yang anda masukkan
tidak valid\n");
        ulang = false;
    }
}
}

```

```

//membuat method yang berfungsi mencari volume dan mencetak
kelayar
private void Volume(){
    this.volume = (this.pi * (this.jari * this.jari) *
this.tinggi * 1) / 3;
    System.out.println("Volume Kerucut = " +
Math.round(this.volume) + " cm\n");
}

```

```

//membuat method berfungsi mencari luas selimut dan
mencetaknya
private void LuasSelimut(){
    this.luas = this.pi * this.jari * this.s;
    System.out.println("Luas Selimut Kerucut = " +
Math.round(this.luas) + " cm\n");
}

```

```

//membuat method mencari luas alas, dan mencetaknya
private void LuasAlas(){
    this.luas = this.pi * this.jari * this.jari;
    System.out.println("Luas Alas Kerucut = " +
Math.round(this.luas) + " cm\n");
}

```

```

//method untuk mencari luas permukaan, dan mencetak
private void LuasPermukaan(){
    this.luas = this.pi * this.jari * (this.jari + this.s);
    System.out.println("Luas Permukaan Kerucut = " +
Math.round(this.luas) + " cm\n");
}

```

```

//method Set untuk variabel jari
private void setJari(double jari){
    this.jari = jari;
}

//method set variabel tinggi
private void setTinggi(double tinggi){
    this.tinggi = tinggi;
}

//method set variabel s(garis pelukis)
private void setGarisPelukis(double s){
    this.s = s;
}

//method mencari nilai garis pelukis dan mengembalikannya
private double getGarisPelukis(){
    this.s = (this.jari*this.jari) +
(this.tinggi*this.tinggi);
    this.s = Math.sqrt(s);
    this.luas = this.pi * this.jari * this.s;
    return this.s;
}

//method menampilkan nama pembuat program
public void show(){
    System.out.println("\nNama\t: Rizqillah\nNIM\t:
1957301020\n");
}
}

Class Main :
public class Main {
    public static void main(String[] args) {
        //membuat objek dari class KerucutRZQ
        KerucutRZQ Rizqillah = new KerucutRZQ();
    }
}

```

## Hasil :

```
Nama      : Rizqillah
NIM       : 1957301020

Kerucut
Anda ingin Menghitung?
1.Volume
2.Luas Selimut
3.Luas Alas
4.Luas Permukaan
5.Cari Nilai Garis Pelukis

Masukkan Input anda : 5

Masukkan jumlah nilai Jari-jari (cm)      : 5
Masukkan jumlah nilai Tinggi (cm)         : 12
Nilai Garis Pelukis adalah = 13 cm

Anda ingin Keluar?? (Y/N)y

=====

BUILD SUCCESSFUL (total time: 16 seconds)
```

## Penjelasan :

Pada program kerucut diatas, saya membuat sebuah program yang menggunakan konsep OOP dengan cara membuat class bernama KerucutRZQ, kemudian saya deklarasi beberapa variabel global yang diperlukan, dan pada class tersebut saya menggunakan beberapa method, yaitu method construct saya gunakan untuk membuat program bisa berjalan tanpa perlu memanggil method di class main.

Diawal kemunculan hasil program, akan bertanya kita akan menghitung masalah apa, kemudian mengeksekusi inputan kita, dan program akan berjalan secara bertahap, kemudian program akan meminta beberapa inputan dari user yang kemudian akan digunakan untuk perhitungan pada Kerucut. Dan adapun saya menggunakan beberapa method agar perhitungan yang dilakukan terjadi pada method dan langsung dicetak hasil. Dan ketika program telah berjalan sampai akhir, kemudian program akan bertanya apakah kita akan keluar atau tidak, jika kita menjawab tidak, maka program akan terlooping kebagian awal diminta inputan pemilihan. Dan jika menjawab iya, maka program akan selesai.

### 2.2.2 Bangun Datar

#### ▪ Lingkaran

##### Coding :

```
package bangunDatar;
import java.util.Scanner; //Library Scanner

public class LingkaranRZQ {
    //Atribut yang dibutuhkan, dan untuk pi nilainya tetap, dan
    //lainnya bersifat private
    private final double pi = 3.14;
    private double jari, luas, keliling;
    Scanner input = new Scanner(System.in); //deklarasi
    variabel scanner yang akan digunakan

    //membuat method construct
    public LingkaranRZQ(){
        show();
        //variabel method construct
        int pilihan;
        boolean ulang = true;

        //melakukan perulangan jika nilai ulang masih true
        while(ulang){
            //melakukan inputan pilihan
            System.out.print("Lingkaran\nAnda ingin
Menghitung?\n1.Luas\n2.Keliling\nMasukkan Input anda : ");
            pilihan = input.nextInt();

            //mengecek apakah pilihan yang dipilih antara 1
atau 2
            if(pilihan == 1 || pilihan == 2){

                //jika pilihan benar, maka akan diminta input
jari-jari yang diketahui dan akan memanggil method setJari
                //untuk diisi nilai jari-jari melalui parameter
                System.out.print("Masukkan jumlah nilai jari-
jari (cm) : ");
                this.jari = input.nextDouble();
                setJari(this.jari);

                //apabila pilihan yang dimasukkan sama dengan 1
maka akan dipanggil method getLuas, agar dicari berapa luasnya
```

```

        if(pilihan == 1){
            getLuas();
            //dan jika selain pilihan 1, maka akan
dipanggil method getKeliling agar dicari nilai kelilingnya.
            //alasan saya disini mengapa tidak memakai else
if,
            //adalah dikarenakan sebelumnya sudah dicek
bahwa hanya pilihan 1 dan 2 yang bisa masuk kesini
        } else {
            getKeliling();
        }

        //pada sintaks dibawah ini untuk bertanya
kepada user ingin keluar atau tidak, pilihan yang bisa
dimasukkan
        //hanyalah huruf (y,Y,n,N), selain dari itu
akan muncul tidak valid
        System.out.print("Anda ingin Keluar?? (Y/N)");
        String result = input.next();

        //mengecek apakah yang diinput Y atau y
        //maka akan menghasilkan nilai ulang = false,
yang berarti keluar dari while
        switch (result) {
            case "Y":
            case "y":
            case "yes":
            case "Yes":
                ulang = false;
                break;

            //mengecek apakah yang diinput N atau n
            //maka akan menghasilkan nilai ulang = true,
dan berarti perulangan while akan tetap berjalan
            case "N":
            case "n":
            case "no":
            case "No":
                ulang = true;
                break;

            //dan jika yang dimasukkan adalah lainnya,
maka akan mencetak Inputan SALAH!!

```

```

        //dan memberi nilai false pada variabel
ulang, agar perulangan berhenti
        default:
            System.out.println("Inputan SALAH!!");
            ulang = false;
            break;
    }

    System.out.print("\n=====
    ===\n\n");

        //dan jika pilihan yang dimasukkan bukan angka 1
atau 2, maka akan mencetak pilihan anda tidak valid
        //dan memberi nilai false pada variabel ulang, agar
keluar dari while
    } else {
        System.out.print("Pilihan yang anda masukkan
tidak valid\n");
        ulang = false;
    }
}

//membuat method untuk mencari hasil nilai luas
private void getLuas(){
    this.luas = this.pi * this.jari * this.jari;
    System.out.println("luas lingkaran " +
Math.round(this.luas) + " cm\n");
}

//membuat method untuk mencari hasil nilai keliling
private void getKeliling(){
    this.keliling = this.pi * 2 * this.jari;
    System.out.println("keliling lingkaran : " +
Math.round(this.keliling) + " cm\n");
}

//membuat method untuk menerima nilai jari-jari pada
variabel global

```

```

        private void setJari(double jari) {
            this.jari = jari;
        }

        //method menampilkan nama pembuat program
        public void show(){
            System.out.println("\nNama\t: Rizqillah\nNIM\t:
1957301020\n");
        }
    }

Class Main :
public class Main {
    public static void main(String[] args) {
        //membuat objek dari class LingkaranRZQ
        LingkaranRZQ Rizqillah = new LingkaranRZQ();
    }
}

```

#### Hasil :

```

Nama      : Rizqillah
NIM       : 1957301020

Lingkaran
Anda ingin Menghitung?
1.Luas
2.Keliling
Masukkan Input anda : 1
Masukkan jumlah nilai jari-jari (cm) : 5
luas lingkaran 79 cm

Anda ingin Keluar?? (Y/N)y

=====

BUILD SUCCESSFUL (total time: 10 seconds)

```

**Penjelasan :**

Pada program lingkaran diatas, saya membuatnya menggunakan konsep OOP yang telah dipelajari. Yaitu pertama saya membuat sebuah class bernama LingkaranRZQ, kemudian saya memberi beberapa atribut(variabel global) pada class tersebut. Dan pada class tersebut, saya menggunakan method construct agar program berjalan dengan sendirinya ketika objek telah dibuat. Kemudian diawal program muncul, akan menampilkan beberapa informasi beserta pilihan yang bisa kita pilih jika kita ingin mencari/menghitung suatu hal dari lingkaran.

Ketika kita selesai memilih, maka program akan meminta inputan nilai-nilai yang dibutuhkan, yang nantinya akan digunakan sebagai nilai pencari terhadap pilihan si user. Nilai pencarian tersebut didapatkan dari dalam method yang khusus digunakan untuk mencari hasil kemudian menampilkannya kelayar agar si user bisa mengetahui hasil dari pencarian yang dilakukannya. Setelah hasil diketahui dan dicetak kelayar, maka program akan bertanya ke user, apakah dia akan keluar, ataukah tidak. Jika iya maka program akan berhenti, dan jika iya maka program akan mengulang ketempat inputan pilihan yang akan dilakukan.



### **BAB 3**

### **KESIMPULAN**

Dapat kita simpulkan, bahwa dalam membuat program, OOP(Object Oriented Programming) sangat dibutuhkan, agar program yang kita buat bisa menjadi lebih simple, elegan, mudah dimodifikasi, dan lain-lain.

Untuk membuat program agar bisa dikatakan telah memiliki OOP adalah dengan adanya beberapa dari konsep Class, Object, attribute, Method, Construct, Inheritance, Polymorphism, Encapsulation, dan Abstract.

Pada pembuatan suatu program yang mungkin membutuhkan hal yang sangat kompleks. Maka konsep OOP tersebut sangatlah penting dipelajari. Dan dengan adanya OOP pada pemrograman, membuat suatu program menjadi lebih mudah dan powerful.

## DAFTAR PUSTAKA

- 1) DuniaMatematika. 5 April 2014. Jenis-jenis Bangun Ruang dan Rumusnya. <https://duniamatematika.com/>. Diakses pada 26 Februari 2021.
- 2) Advernesia. 8 Rumus Bangun Datar. <https://www.advernesia.com/blog/matematika/bangun-datar/>. Diakses pada 26 Februari 2021.
- 3) Agustian. 29 November 2020. Bangun Datar : Pengertian, Macam, Sifat, dan Rumusnya. <https://rumuspintar.com/bangun-datar/>. Diakses pada 26 Februari 2021.
- 4) Guru Billy. 30 Januari 2021. Rumus Kerucut | Pengertian, Rumus Volume, Rumus Luas dan Contoh Soal. <https://gurusekolah.co.id/>. Diakses pada 26 Februari 2021.
- 5) Saintif. Rumus Luas Lingkaran. <https://saintif.com/>. Diakses pada 26 Februari 2021.
- 6) Aimyaya. Luas Permukaan Kerucut. <https://www.cara.aimyaya.com/>. Diakses pada 27 Februari 2021.
- 7) Kelas Pintar. 12 Oktober 2020. Rumus Volume Kerucut. <https://www.kelaspintar.id/>. Diakses pada 27 Februari 2021.
- 8) Azzahra Rahma. 16 Februari 2020. Rumus Luas Permukaan Kerucut dan Rumus Volume beserta Contohnya. <https://rumus.co.id/luas-permukaan-kerucut/>. Diakses pada 27 Februari 2021.